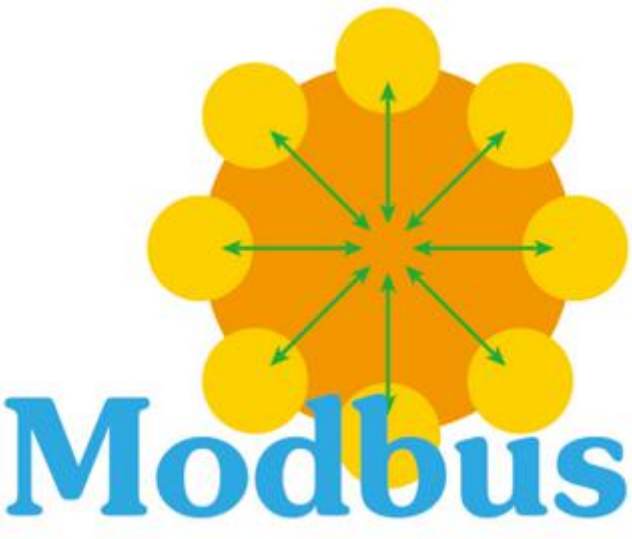
MODBUS 协议详解



目 录

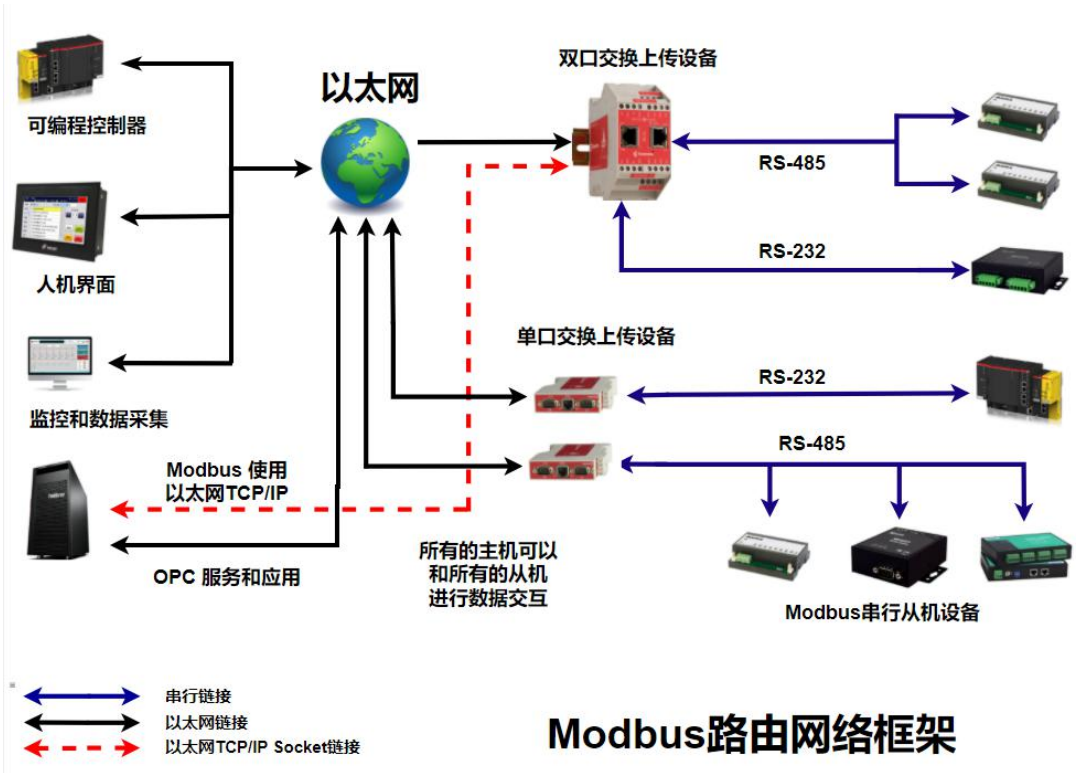
MODBUS 协议详解……1
1. Modbus 简介……3
2. Modbus 功能码说明……4
2.1. 数据模型说明……4
2.2. 寄存器和地址……4
2.3. 功能码说明……5
2.4. PLC 地址和协议地址区别……5
2.5. 寄存器 PLC 地址……6
2.6. 寄存器协议地址……6
3. Modbus 指令说明……6
3.1. 读线圈寄存器 01H……6
3.2. 读离散输入寄存器 02H……7
3.3. 读保持寄存器 03H……9
3.4. 读输入寄存器 04H……10
3.5. 写单个线圈寄存器 05H……12
3.6. 写单个保持寄存器 06H……13
3.7. 写多个线圈寄存器 0FH……14
3.8. 写多个保持寄存器 10H……15
4. Modbus 自学资料……17

1. Modbus 简介

Modbus 是一种串行通信协议，是 Modicon 公司（现在的施耐德电气 Schneider Electric）于 1979 年为使用可编程逻辑控制器（PLC）通信而发表。Modbus 已经成为工业领域通信协议事实上的业界标准，并且现在是工业电子设备之间常用的连接方式。[1] Modbus 比其他通信协议使用的更广泛的主要原因有：

1. 公开发表并且无著作权要求
2. 易于部署和维护
3. 对供应商来说，修改移动本地的比特或字节没有很多限制

Modbus 允许多个（大约 240 个）设备连接在同一个网络上进行通信，举个例子，一个由测量温度和湿度的设备，并且将结果发送给计算机。在数据采集与监视控制系统（SCADA）中，Modbus 通常用来连接监控计算机和远程终端控制系统（RTU）。



2. Modbus 功能码说明

2.1. 数据模型说明

数据模型	对象类型	操作	对象类型
线圈（Coils）	位操作	读写	这种类型的数据可以由应用程序更改程序。
离散输入（Discretes input）	16 位字操作	只读	这种类型的数据可以由 I/O 系统提供
保存寄存器（Hodling Registers）	16 位字操作	读写	这种类型的数据可以由应用程序更改程序。
输入寄存器（input Registers）	位操作	只读	这种类型的数据可以由 I/O 系统提供

例：

线圈（DO）：电磁阀输出，MOSFET 输出，LED 显示等。

离散输入（DI）：拨码开关，类开关等。

保存寄存器（AO）：模拟量输出值，数据参数，变量阀输出大小，传感器数据。

输入寄存器（AI）：模拟量输入值，输入性传感器。

2.2. 寄存器和地址

功能码	说明	寄存器 PLC 地址	协议地址	操作类型
01（0x01）	读取单个线圈	00001-09999	0000H-FFFFH	单个或多个
02（0x02）	读离散输入	10001-19999	0000H-FFFFH	单个或多个
03（0x03）	读保存寄存器	40001-49999	0000H-FFFFH	单个或多个
04（0x04）	读输入寄存器	30001-39999	0000H-FFFFH	单个或多个
05（0x05）	写单个线圈	00001-09999	0000H-FFFFH	单个
06（0x06）	写单个寄存器	40001-49999	0000H-FFFFH	单个
15（0x0F）	写多个线圈	00001-09999	0000H-FFFFH	多个

16（0x10）	写多个寄存器	40001-49999	0000H-FFFFH	多个

表 1 MODBUS 部分功能码说明

注：详细功能码说明 Modbus_Application_Protocol_V1_1b3 (page 11)

Sub-function code		Name
Hex	Dec	
00	00	Return Query Data
01	01	Restart Communications Option
02	02	Return Diagnostic Register
03	03	Change ASCII Input Delimiter
04	04	Force Listen Only Mode
	05.. 09	RESERVED
0A	10	Clear Counters and Diagnostic Register
0B	11	Return Bus Message Count
0C	12	Return Bus Communication Error Count
0D	13	Return Bus Exception Error Count
0E	14	Return Server Message Count
0F	15	Return Server No Response Count
10	16	Return Server NAK Count
11	17	Return Server Busy Count
12	18	Return Bus Character Overrun Count
13	19	RESERVED
14	20	Clear Overrun Counter and Flag
N.A.	21 ... 65535	RESERVED

2.3. 功能码说明

功能码可以分为位操作和字操作两类。位操作的最小单位为 BIT，字操作的最小单位为两个字节。

【位操作指令】 读线圈状态 01H，读(离散)输入状态 02H，写单个线圈 06H 和写多个线圈 0FH。

【字操作指令】 读保持寄存器 03H，写单个寄存器 06H，写多个保持寄存器 10H。

2.4. PLC 地址和协议地址区别

PLC 地址可以理解为协议地址的变种，在触摸屏和 PLC 编程中应用较为广泛。

2.5. 寄存器 PLC 地址

寄存器 PLC 地址指存放于控制器中的地址，这些控制器可以是 PLC，也可以使触摸屏，或是文本显示器。PLC 地址一般采用 10 进制描述，共有 5 位，其中第一位代码寄存器类型。第一位数字和寄存器类型的对应关系如表 1 所示。PLC 地址例如 40001、30002 等。

2.6. 寄存器协议地址

寄存器协议地址指指通信时使用的寄存器地址，例如 PLC 地址 40001 对应寻址地址 0x0000，40002 对应寻址地址 0x0001，寄存器寻址地址一般使用 16 进制描述。再如，PLC 寄存器地址 40003 对应协议地址 0002，PLC 寄存器地址 30003 对应协议地址 0002，虽然两个 PLC 寄存器寄存器通信时使用相同的地址，但是需要使用不同的命令访问，所以访问时不存在冲突。

3. Modbus 指令说明

3.1. 读线圈寄存器 01H

1) 描述:
读 MODBUS 从机线圈寄存器当前状态。
2) 查询:
例如从机地址为 11H，线圈寄存器的起始地址为 0013H，结束地址为 0037H。该次查询总共访问 37 个线圈寄存器。
表读线圈寄存器 - 查询

说明	HEX
从机地址	11
功能码	01
寄存器起始地址高字节	00
寄存器起始地址低字节	13
寄存器数量高字节	00

寄存器数量低字节	25
CRC 校验高字节	0E
CRC 校验低字节	84

3) 响应

响应负载中的各线圈状态与数据内容每位相对应。1 代表 ON，0 代表 OFF。若返回的线圈数不为 8 的倍数，则在最后数据字节未尾使用 0 代替。

表读线圈寄存器 - 响应

说明	HEX
从机地址	11
功能码	01
返回字节数	05
数据 1	CD
数据 2	6B
数据 3	B2
数据 4	0E
数据 5	1B
CRC 校验高字节	45
CRC 校验低字节	E6

数据1是线圈0013H到线圈001AH的状态为CDH，二进制值为11001101，该字节的最高字节为线圈 001AH，最低字节为线圈 0013H。线圈 001AH 到线圈 0013H 的状态分别为 ON-ON-OFF-OFF-ON-ON-OFF-ON。

3.2. 读离散输入寄存器 02H

1) 说明

读离散输入寄存器状态。

2) 查询

从机地址为 11H。离散输入寄存器的起始地址为 00C4H，结束寄存器地址为 00D9H。总共访问 32 个离散输入寄存器。

表读离散输入寄存器——查询

说明	HEX
从机地址	11
功能码	02
寄存器起始地址高字节	00
寄存器起始地址低字节	C4
寄存器数量高字节	00
寄存器数量低字节	16
CRC 校验高字节	BA
CRC 校验低字节	A9

3) 响应

响应各离散输入寄存器状态，分别对应数据区中的每位值，1 代表 ON；0 代表 OFF。第一个数据字节的 LSB(最低字节)为查询的寻址地址，其他输入口按顺序在该字节中由低字节向高字节排列，直到填充满 8 位。下一个字节中的 8 个输入位也是从低字节到高字节排列。若返回的输入位数不是 8 的倍数，则在最后的数据字节中的剩余位至该字节的最高位使用 0 填充。

表读输入寄存器 - 响应

说明	HEX
从机地址	11
功能码	02
返回字节数	03
数据 1	AC

数据 2	DB
数据 4	35
CRC 校验高字节	20
CRC 校验低字节	18

离散输入寄存器 00D4H 到 00D9H 的状态为 35H (二进制 00110101)。输入寄存器00D9H为左数第3位,输入寄存器00D4为最低位,输入寄存器00D9H到00D4H的状态分别为ON-ON-OFF-ON-OFF-ON。00DBH寄存器和00DAH寄存器被 0 填充。

3.3. 读保持寄存器 03H

1) 说明

读保持寄存器。可读取单个或多个保持寄存器。

2) 查询

从机地址为 11H。保持寄存器的起始地址为 006BH，结束地址为 006DH。该次查询总共访问 3 个保持寄存器。

表读保持寄存器-查询

说明	HEX
从机地址	11
功能码	03
寄存器起始地址高字节	00
寄存器起始地址低字节	6B
寄存器数量高字节	00
寄存器数量低字节	03
CRC 校验高字节	76
CRC 校验低字节	87

3) 响应

保持寄存器的长度为 2 个字节。对于单个保持寄存器而言，寄存器高字节数据先被传输，低字节数据后被传输。保持寄存器之间，低地址寄存器先被传输，高地址寄存器后被传输。

表读保持寄存器-响应

说明	HEX
从机地址	11
功能码	03
返回字节数	06
数据 1 高字节	00
数据 1 低字节	6B
数据 2 高字节	00
数据 2 低字节	13
数据 3 高字节	00
数据 3 低字节	00
CRC 校验高字节	38
CRC 校验低字节	B9

3.4. 读输入寄存器 04H

1) 说明

读输入寄存器命令。该命令支持单个寄存器访问也支持多个寄存器访问。

2) 查询

从机地址为 11H。输入寄存器的起始地址为 0008H，寄存器的结束地址为 0009H。本次访问访问 2 个输入寄存器。

表读输入寄存器-查询

说明	HEX
从机地址	11
功能码	04
寄存器起始地址高字节	00
寄存器起始地址低字节	08
寄存器数量高字节	00
寄存器数量低字节	02
CRC 校验高字节	F2
CRC 校验低字节	99

3) 响应

输入寄存器长度为 2 个字节。对于单个输入寄存器而言，寄存器高字节数据先被传输，低字节数据后被传输。输入寄存器之间，低地址寄存器先被传输，高地址寄存器后被传输。

表读寄存器-响应

说明	HEX
从机地址	11
功能码	04
返回字节数	04
数据 1 高字节	00
数据 1 低字节	0A
数据 2 高字节	00
数据 2 低字节	0B
CRC 校验高字节	8B
CRC 校验低字节	80

3.5. 写单个线圈寄存器 05H

1) 说明

写单个线圈寄存器。FF00H 值请求线圈处于 ON 状态，0000H 值请求线圈处于 OFF 状态。05H 指令设置单个线圈的状态，15H 指令可以设置多个线圈的状态，两个指令虽然都设定线圈的 ON/OFF 状态，但是 ON/OFF 的表达方式却不同。

2) 查询

从机地址为 11H，线圈寄存器的地址为 00ACH。使 00ACH 线圈处于 ON 状态，即数据内容为 FF00H。

表写单个线圈-查询

说明	HEX
从机地址	11
功能码	05
寄存器起始地址高字节	00
寄存器起始地址低字节	AC
数据 1 高字节	FF
数据 1 低字节	00
CRC 校验高字节	4E
CRC 校验低字节	8B

3) 响应

2.5.1 强制单个线圈——响应

说明	HEX
从机地址	11
功能码	05
寄存器地址高字节	00
寄存器地址低字节	AC
寄存器 1 高字节	FF

寄存器 1 低字节	00
CRC 校验高字节	4E
CRC 校验低字节	8B

3.6. 写单个保持寄存器 06H

1) 说明

写保持寄存器。注意 06 指令只能操作单个保持寄存器，16 指令可以设置单个或多个保持寄存器。

2) 查询

从机地址为 11H。保持寄存器地址为 0001H。寄存器内容为 0003H。

表 2.6.1 写单个保持寄存器——查询

说明	HEX
从机地址	11
功能码	06
寄存器起始地址高字节	00
寄存器起始地址低字节	01
数据 1 高字节	00
数据 1 低字节	03
CRC 校验高字节	9A
CRC 校验低字节	9B

3) 响应

表 2.6.2 写单个保持寄存器——响应

说明	HEX
从机地址	11

功能码	06
寄存器起始地址高字节	00
寄存器起始地址低字节	01
数据 1 高字节	00
数据 1 低字节	01
CRC 校验高字节	1B
CRC 校验低字节	5A

3.7. 写多个线圈寄存器 0FH

1) 说明

写多个线圈寄存器。若数据区的某位值为“1”表示被请求的相应线圈状态为 ON，若某位值为“0”，则为状态为 OFF。

2) 查询

从机地址为 11H，线圈寄存器的起始地址为 0013H，线圈寄存器的结束地址为 001CH。总共访问 10 个寄存器。寄存器内容如下表所示。

传输的第一个字节 CDH 对应线圈为 0013H 到 001AH，LSB（最低位）对应线圈 0013H，传输第二个字节为 01H，对应的线圈为 001BH 到 001CH，LSB 对应线圈 001CH，其余未使用位使用 0 填充。

表写多个线圈寄存器——查询

说明	HEX
从机地址	11
功能码	0F
寄存器起始地址高字节	00
寄存器起始地址低字节	13

寄存器数量高字节	00
寄存器数量低字节	0A
字节数	02
数据 1	CD
数据 2	01
CRC 校验高字节	BF
CRC 校验低字节	0B

3) 响应

表 2.7.1 写多个线圈寄存器——响应

说明	HEX
从机地址	11
功能码	0F
寄存器起始地址高字节	00
寄存器起始地址低字节	13
寄存器数量高字节	00
寄存器数量低字节	0A
字节数	02
CRC 校验高字节	99
CRC 校验低字节	1B

3.8. 写多个保持寄存器 10H

1) 说明

写多个保持寄存器。

2) 查询

从机地址为 11H。保持寄存器的其实地址为 0001H，寄存器的结束地址为 0002H。总共访问 2 个寄存器。保持寄存器 0001H 的内容为 000AH，保持寄存器 0002H 的内容为 0102H。

表写多个保持寄存器——请求

说明	HEX
从机地址	11
功能码	10
寄存器起始地址高字节	00
寄存器起始地址低字节	01
寄存器数量高字节	00
寄存器数量低字节	02
字节数	04
数据 1 高字节	00
数据 1 低字节	0A
数据 2 高字节	01
数据 2 低字节	02
CRC 校验高字节	C6
CRC 校验低字节	F0

3) 响应

表 2.8.3 写多个保持寄存器——响应

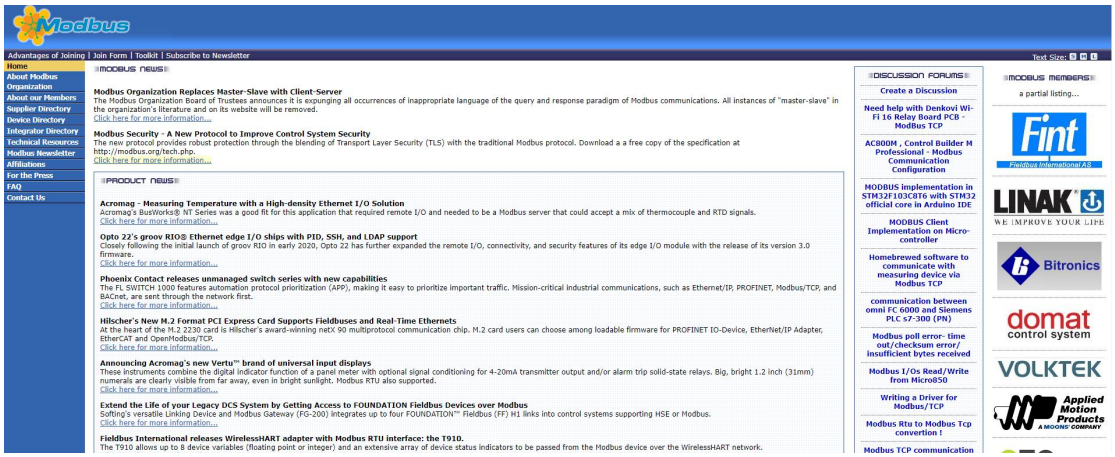
说明	HEX
从机地址	11
功能码	10

寄存器起始地址高字节	00
寄存器起始地址低字节	01
寄存器数量高字节	00
寄存器数量低字节	02
CRC 校验高字节	12
CRC 校验低字节	98

4. Modbus 自学资料

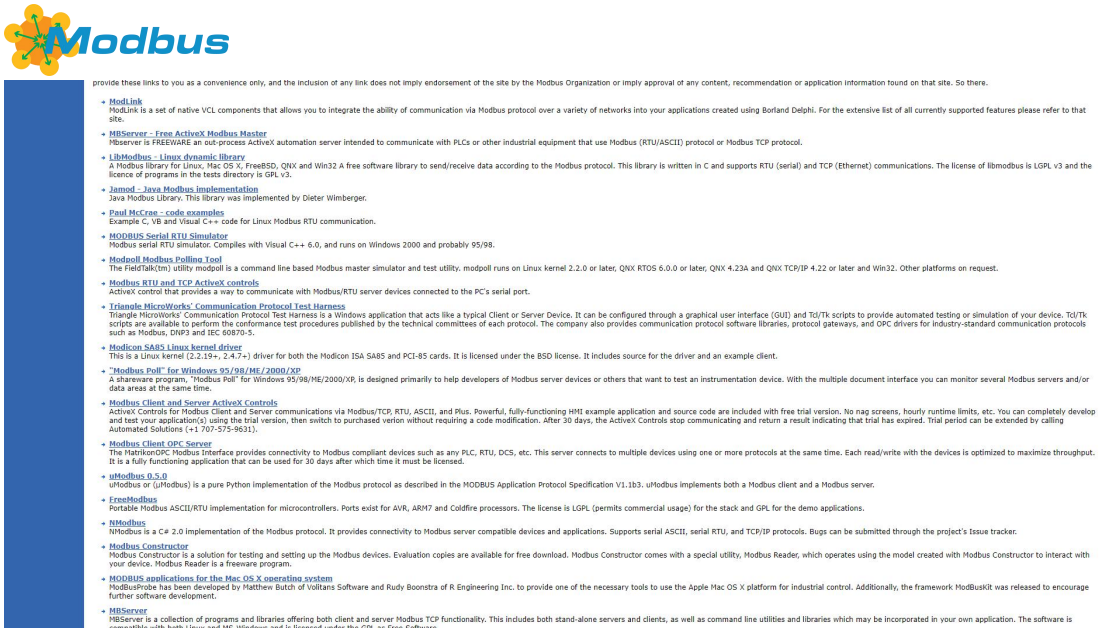
4.1.Modbus 官网

https://modbus.org/



其下有一个页面：https://modbus.org/tech.php ，有很多 modbus 相关的工具，不同编程语言下 modbus 的资源，可以根据需求查询相关资料。

各大开源网站，Github，谷歌等等，有大量的案例开源参考学习。



provide these links to you as a convenience only, and the inclusion of any link does not imply endorsement of the site by the Modbus Organization or imply approval of any content, recommendation or application information found on that site. So there.

- **ModLink**
  ModLink is a set of native VCL components that allows you to integrate the ability of communication via Modbus protocol over a variety of networks into your applications created using Borland Delphi. For the extensive list of all currently supported features please refer to that site.
- **MBServer - Free ActiveX Modbus Master**
  Mbserver is FREEWARE an out-process ActiveX automation server intended to communicate with PLCs or other industrial equipment that use Modbus (RTU/ASCII) protocol or Modbus TCP protocol.
- **LibModbus - Linux dynamic library**
  A Modbus library for Linux, Mac OS X, FreeBSD, QNX and Win32 A free software library to send/receive data according to the Modbus protocol. This library is written in C and supports RTU (serial) and TCP (Ethernet) communications. The license of libmodbus is LGPL v3 and the licence of programs in the tests directory is GPL v3.
- **Jamod - Java Modbus implementation**
  Java Modbus Library. This library was implemented by Dieter Wimberger.
- **Paul McCrae - code examples**
  Example C, VB and Visual C++ code for Linux Modbus RTU communication.
- **MODBUS Serial RTU Simulator**
  Modbus serial RTU simulator. Compiles with Visual C++ 6.0, and runs on Windows 2000 and probably 95/98.
- **Modpoll Modbus Polling Tool**
  The FieldTalk(tm) utility modpoll is a command line based Modbus master simulator and test utility. modpoll runs on Linux kernel 2.2.0 or later, QNX RTOS 6.0.0 or later, QNX 4.23A and QNX TCP/IP 4.22 or later and Win32. Other platforms on request.
- **Modbus RTU and TCP ActiveX controls**
  ActiveX control that provides a way to communicate with Modbus/RTU server devices connected to the PC's serial port.
- **Triangle MicroWorks' Communication Protocol Test Harness**
  Triangle MicroWorks' Communication Protocol Test Harness is a Windows application that acts like a typical Client or Server Device. It can be configured through a graphical user interface (GUI) and Tcl/Tk scripts to provide automated testing or simulation of your device. Tcl/Tk scripts are available to perform the conformance test procedures published by the technical committees of each protocol. The company also provides communication protocol software libraries, protocol gateways, and OPC drivers for industry-standard communication protocols such as Modbus, DNP3 and IEC 60870-5.
- **Modicon SA85 Linux kernel driver**
  This is a Linux kernel (2.2.19+, 2.4.7+) driver for both the Modicon ISA SA85 and PCI-85 cards. It is licensed under the BSD license. It includes source for the driver and an example client.
- **"Modbus Poll" for Windows 95/98/ME/2000/XP**
  A shareware program, "Modbus Poll" for Windows 95/98/ME/2000/XP, is designed primarily to help developers of Modbus server devices or others that want to test an instrumentation device. With the multiple document interface you can monitor several Modbus servers and/or data areas at the same time.
- **Modbus Client and Server ActiveX Controls**
  ActiveX Controls for Modbus Client and Server communications via Modbus/TCP, RTU, ASCII, and Plus. Powerful, fully-functioning HMI example application and source code are included with free trial version. No nag screens, hourly runtime limits, etc. You can completely develop and test your application(s) using the trial version, then switch to purchased verion without requiring a code modification. After 30 days, the ActiveX Controls stop communicating and return a result indicating that trial has expired. Trial period can be extended by calling Automated Solutions (+1 707-575-9631).
- **Modbus Client OPC Server**
  The MatrikonOPC Modbus Interface provides connectivity to Modbus compliant devices such as any PLC, RTU, DCS, etc. This server connects to multiple devices using one or more protocols at the same time. Each read/write with the devices is optimized to maximize throughput. It is a fully functioning application that can be used for 30 days after which time it must be licensed.
- **uModbus 0.5.0**
  uModbus or (µModbus) is a pure Python implementation of the Modbus protocol as described in the MODBUS Application Protocol Specification V1.1b3. uModbus implements both a Modbus client and a Modbus server.
- **FreeModbus**
  Portable Modbus ASCII/RTU implementation for microcontrollers. Ports exist for AVR, ARM7 and Coldfire processors. The license is LGPL (permits commercial usage) for the stack and GPL for the demo applications.
- **NModbus**
  NModbus is a C# 2.0 implementation of the Modbus protocol. It provides connectivity to Modbus server compatible devices and applications. Supports serial ASCII, serial RTU, and TCP/IP protocols. Bugs can be submitted through the project's Issue tracker.
- **Modbus Constructor**
  Modbus Constructor is a solution for testing and setting up the Modbus devices. Evaluation copies are available for free download. Modbus Constructor comes with a special utility, Modbus Reader, which operates using the model created with Modbus Constructor to interact with your device. Modbus Reader is a freeware program.
- **MODBUS applications for the Mac OS X operating system**
  ModBusProbe has been developed by Matthew Butch of Volitans Software and Rudy Boonstra of R Engineering Inc. to provide one of the necessary tools to use the Apple Mac OS X platform for industrial control. Additionally, the framework ModBusKit was released to encourage further software development.
- **MBServer**
  MBServer is a collection of programs and libraries offering both client and server Modbus TCP functionality. This includes both stand-alone servers and clients, as well as command line utilities and libraries which may be incorporated in your own application. The software is

4.2. 相关文献

Modbus_Application_Protocol_V1_1b3

Modbus_Messaging_Implementation_Guide_V1_0b

Modbus_over_serial_line_V1_02

MB-TCP-Security-v21_2018-07-24

PI_MBUS_300